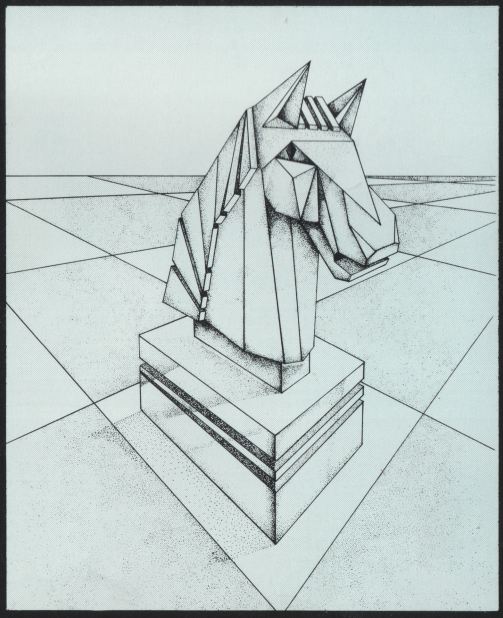
APPLE
II SERIES

COLOSSUS CHESS IV



firebird

USERMANUAL

COLOSSUS CHESS IV

by Martin Bryant

The best chess playing program, for home computers.

© FIREBIRD LICENSEES INC 1986

Table of Contents

SECTION	SUBJECT
1	**INTRODUCTION**
a	General
b	Hardware requirements
c	Running the program
d	Quick guide
2	**DISPLAY**
a	General
b	The board
c	Messages & questions
d	The move record
i	The player names
ii	The clocks
iii	The moves
e	Technical information
i	Lookahead
ii	Positions examined
iii	Best line
iv	Current line
v	Assumed move
3	**PLAYING A GAME**
a	New game
b	Your move
c	Colossus move
d	Game over
4	**KEYBOARD ENTRIES**
a	Cursor movement
b	Numeric data - *IMPORTANT* - PLEASE READ
c	Commands
5	**PROGRAM INFORMATION**
a	Statistics
b	Openings book
c	Results
6	**GAMES & PROBLEMS**
a	Games
b	Problems

SECTION 1 - INTRODUCTION

1.a - General

Colossus is the best computer chess program available on any home computer. It is written using the very latest techniques by a computer chess programmer of eight years experience. It has been tested against a wide range of other chess programs and proven stronger than any of the opposition. It has the widest range of true features and sub-features available, including some, never before implemented on any home computer chess program. Colossus also has a perfect understanding of all the rules of chess, including underpromotions, the fifty move rule and all draws by repetition. It can also handle all the standard mates including King and two Bishops versus King and the very difficult King, Bishop and Knight versus King.

1.b - Hardware requirements

Colossus will run on any Apple II computer with at least 48K of RAM memory and one disk drive. If your computer has 64K RAM memory then two extra facilities and provided in the 'Quantify' command (see later) to allow selection of the 3D graphics and adjustment of the programs "draw" score.

1.c - Running the program

Place the disc in the drive and boot it, either by powering-up or entering the DOS command "PR#6". The program will then run automatically.

1.d - Quick guide

Colossus has so many features, that at first you may be overwhelmed by the sheer size of the manual. However you do not have to read it from cover to cover but merely sample the relevant sections as your knowledge of the program increases.

The only sections you should need to read to get started (apart from this introduction of course) are "3.b - Your Move" and "4.a - Cursor Movement". Then the most commonly used commands in section 4.c are "Go", "New-Game" and "Type".

After that, the other sections can be read at your leisure, as you discover the truly amazing features that COLOSSUS has to offer.

SECTION 2 - DISPLAY

2.a - General

The display uses two screens to provide clear, extensive information as to the state of the game, the programs current thoughts and the past move record. The main screen shows a graphics chess board and various messages and questions as required. The second screen shows the past move record, including the player names and chess clocks and also information on the programs

thought processes including lookahead, positions examined, current-line and best-line with evaluation. The display can be toggled between the two screens by pressing the <SPACE> bar.

2.b - The board

The main screen has a graphics board printout showing the current position. Letters and numbers around the board indicate the algebraic notation used for each square.

2.c - Messages and questions

Various messages and questions are displayed around the board as necessary. These are explained in more detail in later sections.

2.d - The move record

The secondary screen shows the last six moves made by each side in two columns including move numbers. Above the moves are displayed the colours, player names and elapsed-time chess clocks for each side.

2.d.i - The player names

Above each column is displayed the names of the white and black players. The programs name is displayed as 'Colossus' while its opponents name is displayed as 'Opponent'.

2.d.ii - The clocks

Below the player names are displayed the elapsed-time clocks for white and black. The clocks are in the format 'hh:mm:ss' (hh=hours, mm=minutes, ss=seconds). The clocks show the total time used by each side in the game so far. Because the Apple has no hardware timer, the clocks are estimated from the number of positions examined by the program. The accuracy of the clocks may vary depending on certain conditions, but over a long game they should average out correctly. The clock display is only updated after each move is made.

2.d.iii - The moves

The moves are displayed in algebraic notation with the from-square followed by the to-square. The separator between the from- and to- squares indicates whether the move is a capture or not ('x' signifies a capture, '-' signifies a non-capture). Castling moves are indicated with the from- and to- squares of the moving king. En-passant captures are indicated by the letters 'EP' printed after the move.

Promotions are indicated after the move by a '/', followed by a letter to indicate the promoted piece ('N'=knight, 'B'=bishop, 'R'=rook, 'Q'=queen). Checks are indicated by a '+' being printed after the move.

2.e - Technical information

The bottom of the secondary screen is used to display detailed information of the programs thought processes. This information can provide a great insight into how a good chess program works, but only takes a fraction of one percent of the programs thinking time to display it.

2.e.i - Lookahead

When the program thinks about its move, it displays the lookahead depth of its search. The depth is printed in 'ply' or 'half-moves'. The program will search most move sequences to at least this depth, with some being searched much deeper.

2.e.ii - Positions examined

The number of positions examined in the tree of move sequences is displayed. This number is updated at every position and in fact takes very little time because of the single-byte/character type of display. The program examines on average about 300 positions per second in the middlegame.

2.e.iii - Best line

The program displays the best line it has found so far in its search. This can provide you with the usual 'hint' move and also an analysis of the game in the next few moves beyond the 'hint'! (Usually the line will contain a 'null' move. This simply means a passive, non-capturing move has been assumed.) Also displayed is the evaluation of the best-line, as two numbers. The first is the material evaluation (in terms of number of pawns up or down), the second the positional evaluation. A positive number means the program is better, a negative number means the opponent is better. If the best-line is found to lead to a checkmate then the material score will be set to plus or minus 62 with the positional score a measure of how many moves to the checkmate.

2.e.iv - Current line

The move sequence the program is currently considering is displayed. This can be seen to change as the search progresses. The length of the line shown, can be adjusted as required (for further details see the 'Quantify' command described later).

2.e.v - Assumed move

Colossus thinks on its opponents time. It assumes the opponent will make a particular move and then continues thinking about its next move. Sometimes the program will not have a move to assume and then will not think ahead. The move assumed is displayed so that you can follow the programs thought processes and to provide a 'hint' move. However, the quality of this move is

for you to decide. (Note - The opponents clock is only updated if the program is thinking on the opponents time.) If you play a different move from the one assumed, the time is wasted, but the program will correctly predict about one third of your moves.

SECTION 3 - PLAYING A GAME

3.a - New game

When a new game is started, the pieces are set-up in their initial positions, the clocks are reset to '00:00:00', the move display is cleared and you are given the option of moving first. Typing (CTRL)'G' will make the program take the white pieces and move first (for further details see the 'Go' command described later).

3.b. - Your move

Whenever it is your turn to move, the program asks 'Your move?' at the bottom of the main screen. You enter moves by the following cursor positioning method (see the later section 'Cursor movement' for further details).

1) Move the cursor to the square of the piece you wish to move, and press the <RETURN> key. This causes the from-square to be displayed. If you accidentally enter the wrong from-square it can be cancelled by pressing the <ESCAPE> key or the <DELETE> key (APPLE IIe only). (The cursor is shown as an inverse line drawn through the current square).
2) Move the cursor to the square you wish to move to and again press <RETURN>. This causes the 'to-square' to be displayed.

   If the move is illegal, the message 'Illegal' is displayed and the move entry cleared. You must then return to step-1 above and try again.

3) If the move is a pawn promotion the program then asks 'Promote to?'. You must then specify the piece you want to promote to by typing one of the following : 'N'-knight, 'B'-bishop, 'R'-rook or 'Q'-queen. If you press any other key, the program assumes you promote to a queen.

Note: If you wish to enter a castling move, you should move the king two squares either way. If you wish to enter an en-passant capture, you should move the pawn as in a normal capture.

The legal move is now indicated on the board, by flashing the cursor on the from-square, then moving the piece and flashing the cursor on the to-square.

3.c - Colossus move

When the program computes its move it displays the message 'Let me think. . .' below the board. The technical information on the secondary screen is updated as the search progresses. When the program has finished its search, it prints the selected move in the move-record to the right of the board, and also indicates it on the board (in the same way as your moves are indicated). Pressing the <ESCAPE> key, whilst the program is thinking about its move, will cause it to abort its search, display the message 'Escape' and make the best move it has found so far. If the program has found that its move leads to checkmate, it announces the move with the message 'Mate in N' where N is the number of moves till the mate.

3.d - Game over

When a game is completed, the clocks are stopped and the program displays a message, to the right of the board, to indicate the final state of the game. The messages are:-

'Drawn'	-the game has been drawn by 3-fold repetition of position, the 50-move rule or by neither side having enough material to mate the opponent. This occurs in the cases of King, King and Bishop or King and Knight versus a lone King.
'Checkmate'	-the side which moved last delivered checkmate.
'Stalemate'	-the side to move is in stalemate.
'Time-up'	-the side to move lost 'on time'. This occurs only if you are playing an 'All-the-moves' game. (See 'Type' command described later).

The program then asks 'What now?' and waits for you to enter one of the program commands.

Type <CTRL>'N' to start a new game (for further details see the 'New-game' command described later).

SECTION 4 - KEYBOARD ENTRIES

4.a - Cursor movement

The cursor is indicated by an inverse line, drawn through the middle of the current square. It can be moved in two ways.

1) Cursor keys. Four cursor controls, provide single square movement in any direction. The <LEFT-ARROW> and <RIGHT-ARROW> keys give left/right movement, while the <SEMI-COLON> and <SLASH> keys give up/down movement. (The <UP-ARROW> and <DOWN-ARROW> keys also work on APPLE IIe.) If the cursor is moved off one edge of the board, it reappears at the opposite edge.
2) Algebraic keys. The cursor can be moved immediately to any rank, by typing one of the number keys '1' to '8'. It can be moved to any file by typing one of the letters 'A' to 'H'. This allows algebraic notation to be used when entering moves e.g. typing e2<RETURN> e4<RETURN> will enter the common opening move Pe2-e4. Note that if the rank or file of the from- and to- squares is the same, the key sequence can be shortened, i.e. in the above example e2<RETURN>4<RETURN> would also work.

The initial position of the cursor depends on the current side to move.

If white is to move, the cursor starts on square a1.

If black is to move, the cursor starts on square h8.

4.b - Numeric data - *IMPORTANT* - PLEASE READ

Many of the programs commands require numbers to be entered to set-up new values of parameters etc. The program employs a simple to use, error-proof method for entering numbers. First the current value of the parameter is displayed at the query position. To increment the displayed value, use the <SEMI-COLON> OR <UP-ARROW> (APPLE IIe only) key. If the new value is above the upper limit for that parameter then it is reset to its lower limit. To decrement the displayed value, use the <SLASH> OR <DOWN-ARROW> (APPLE IIe only) key. If the new value is below the lower limit for that parameter then it is reset to its upper limit. To enter the new value into the program, press <RETURN>. (If you do not wish to change the current value, then just press <RETURN>, instead of using the "increment" or "decrement" keys.)

4.c - Commands

The following commands can be entered, when it is your turn to move or when the game is over. To enter a command, just type its first letter (shown in capitals in the descriptions below) with the <CTRL> key also pressed. The commands are ordered alphabetically and are designed to be easy and logical to use and to greatly increase your enjoyment of the game of chess.

Note 1: if the selected command requires numeric input, and you are viewing the secondary text screen, then the program will automatically switch back to the primary graphics screen.

Note 2: if a command is entered, then any 'thinking on the opponents time' will be aborted.

Alter-position-Type <CTRL> 'A'
Any legal chess position can be set-up with this command. When selected, the program displays the message:-

'Alter position:SSSSS:Command?' (where SSSSS is the side to move) When in 'Alter-position' mode, a further subset of commands, allow you to adjust the board thus:-

Get initial data: Type 'G'
If after changing part of the position, you decide you've made a mistake, the initial position can be retrieved with this command.

Move-number: Type 'M'
The program asks 'Move number?', thus allowing you to enter the new move number required. (See section 4.b for details on how to enter numeric data.)

Side-to-move: Type 'S'
This toggles the side to move between white and black. When pieces are added to the board, their colour is determined by the current side to move.

Clear,Pawn,Knight,Bishop,Rook,Queen,King: Type 'C', 'P', 'N', 'B', 'R', 'Q' or 'K'. To change a particular square, place the cursor over it and enter one of the above, to determine the new status of that square. (Note that algebraic control of the cursor is not available in 'Alter-position'.) When a square is changed the cursor will move one square to the right to make the setting up of complete rows of pieces very easy.

Wipe: Type 'W'
This removes all pieces from the board, thus allowing positions with very few pieces, to be set-up more quickly.

New-game: Type <CTRL> 'N'
The 'New-game' command is still available when in 'Alter-position' mode. (See full description later.)

Exit: Type 'E'
This allows you to exit from 'Alter-position', when the required position has been achieved. (Remember to set the correct side-to-move before exiting). If the position is illegal for any reason, the message 'Illegal' is displayed and 'Alter-position' will not be exited. You must then correct the error before exiting.

Illegal positions occur with any of the following:

either side has no king

either side has more than one king

the side not to move is in check

either side has a pawn on the first or eighth rank

either side has more than eight pawns or promoted pieces.

When you exit 'Alter-position', if you have made any changes, the current game-record is cleared of all previous moves, so that you cannot 'Back-step' through these (now possibly illegal) moves. (Note that changing the side-to-move or the move-number is also considered as a change.)

Back-step - Type <CTRL> 'B' & **For-step** - Type <CTRL> 'F'
All the moves in the game are stored in the programs memory. This allows you to step backwards or forwards through the game record, to get to any required position. This can be used for several reasons:-

e.g. after you accidentally lose a piece, you can recover your error by using Back-step.

e.g. if you wish to try a different line earlier in the game, you can Back-step to the required position and then carry on playing the different line.

Stepping through the game record only does one move at a time so you must do an even number of steps if you still want to play for the same side.

Note: In fact the program only has room in memory for about the most recent 120 moves. This should be enough to store most games completely.
If however the game goes on for more than 120 moves, the game record is automatically updated to store the most recent 120. You cannot then Back-step right to the beginning of the game.

Choose next best -Type <CTRL> 'C'
After the program has made a move, you can instruct it to choose a different move from those remaining. You can use this repeatedly until all the moves have been used up. When no more moves are available, it prints the message '*NO MORE MOVES*' and starts again choosing from the whole movelist. If the program is still

using its openings bok, it will choose the next book alternative (selected at random). This can be used to examine the programs book choices to select the line you prefer. (No non-book moves are considered!) If the program is out of book, it will choose the move it considers to be its next best choice from the remaining moves.

Disc/Tape - Type <CTRL> 'D'
This allows you to save/load move-records and positions to/from disc. The program asks 'Catalog, Delete, Load or Save?'. Type 'C' to see the disc catalog on the lower half of the secondary screen, type 'D' to delete a file type 'L' to load a previously saved record, or type 'S' if you wish to save the current record. Any other keypress will abort the 'Disc' command. For deleting, loading and saving, the program asks for the file numebr (from 0 to 255), which is used as part of the filename on the disc. (See section 4.b for details on how to enter numeric data.) If any error occurs during loading the current move-record will be lost. Note that the program parameters, such as playing speed, elapsed time, are not saved and so may need to be reset after loading. Any disc errors are printed and the command aborted.

Elapsed-time clocks - Type <CTRL> 'E'
The elapsed time clocks for both sides can be set from 00:00:00 to 255:59:59. First the program asks for the new value for the white clock and then the new value for the black clock. (See section 4.b for details on how to enter numeric data.) This can be used to set-up reasonable values for the elapsed time clocks, if you change the playing speed in mid-game. The program makes extensive use of the elapsed time clocks to determine how quickly it should play e.g. if you increase the elapsed time on the programs clock it will then play quicker to keep within the average time it is currently set to.

Go - Type<CTRL> 'G'
This forces the program to move for the current side to move. This can be used repeatedly to make the program play several moves against itself or to force the program to resume play, for the side to move, after the 'Supervisor' command has been entered (see later). It also allows you to switch sides with the program in mid-game. After a new game has been initiated, this will make the program play for white. In this case the program automatically clears the clocks and sets the orientation for you to play the black pieces up the board.
This command is ignored if the game is over.

Legal moves - Type <CTRL> 'L'
All the legal moves by any piece of the side to move can be shown. First position the cursor on the subject piece and type <CTRL> 'L'. The legal moves will be shown in sequence by flashing the cursor on the target squares. The cursor is then returned to the subject square. If no legal moves exist the cursor will not move. This can be used as a tutorial aid for learners.

New-game - Type <CTRL> 'N'
A new game can be started with this command. If you have not used 'Alter-position' since the last 'New-game', then the game record is left intact until the first move of the new game is entered. This allows you to replay through the whole of the last game, by using the 'For-step' or 'Replay' commands.

Orientation - Type <CTRL> '0'
The orientation of the board printout can be reversed, so that you can play the black pieces up the board. The algebraic notation around the board is also reversed. This is most useful when you wish to play black against the program.

Play-self - Type <CTRL> 'P'
This makes the program play a game against itself, by moving for both sides. When a game is completed, the program pauses for a few seconds, to allow you to study the final position, and then starts a new game automatically. This command is ignored if the program is in 'problem' or 'infinite' mode (see later), or the game is over. Pressing the <ESCAPE> key, whilst the program is thinking about its move (but not while it is indicating its move on the board) or after the game is over, will stop the program playing against itself and allow you to resume play for the current side to move. When playing itself, both player names are displayed as 'Colossus'.

Quantify parameters - Type <CTRL> 'Q'
This allows you to adjust some of the programs internal parameters to make it function in different ways. (See section 4.b for details on how to enter numeric data.) The program first asks 'Book?'. The parameter value can be 0 or 1. If it is set to 0 the program will not search its 'openings book' for moves (effectively making the openings easier for you). If it is set to 1 the program will use its book in the normal manner. The program then asks 'Prediction?'. The parameter value can be 0 or 1. If it is set to 0 the program will not predict your move or think ahead on your time (effectively making the whole game easier for you). If it is set to 1 the program will predict your move in the normal manner. The program then asks 'Line depth?'. The number of moves in the

"current-line" display can be set from 1 to 15 ply. The default is 2, i.e. the moves at the first two ply of the tree are displayed. If you only have 48K RAM in your machine, then the 'Quantitfy' command is now exited. If however you have 64K RAM then the following parameters can also be adjusted. The program then asks 'Dimensions?'. The parameter value can be 2 or 3. If it is set to 2 (the default) the graphics board printout will be the two-dimensional representation as used in chess literature. If it is set to 3 the board is displayed as 3-dimensional as though you were looking at a real board and pieces. (64K RAM MACHINES ONLY.) The program then asks 'Draw score?'. The "material" part can be set from −9 to +9 and the "positional" part from −60 to +60. The lower the draw score is set the more the program will keep trying to win. The higher the draw score is set the more the program will be happy to allow a draw (effectively making it easier for you to achieve a draw by repetition of position). (64K RAM MACHINES ONLY.)

Replay - Type <CTRL> 'R'
The whole game-record can be replayed from the start to the final position to demonstrate the game-so-far to a spectator. The program pauses between moves to allow you to follow the game. The pause time can be set from 0 to 20 seconds. (See section 4.b for details on how to enter numeric data). You can interrupt the replay by pressing the <ESCAPE> key during one of the pauses. When an action replay is in progress, the message '*Replaying*' is displayed below the board.

Supervisor - Type <CTRL> 'S'
This stops the program making any moves, and allows you to play moves for both sides. This allows you to set-up a required position by playing through a sequence of moves, or to play a friend using the programs board display. All moves entered are still checked for legality. When supervising, both player names are displayed as "Opponent".

Type of playing mode - Type <CTRL> 'T'
The program has six modes of play, to allow different speed and style games to be selected. The more time you give the program to think about its move, the better it will play. The program asks 'Type?', to allow you to select one of the following:- (See section 4.b for details on how to enter numeric data.)

1) Tournament mode
   Full tournament parameters can be set-up.
   First, the program asks for the move number of the first time control.

Second, the program asks for the number of moves to secondary time controls.
Third, the program asks for the time of the first time control.
Fourth, the program asks for the time to the secondary time controls.
The program is not strict about your time control and will not claim a win 'on time', if you exceed any time control. The program will usually never exceed its own time control.

2) Average mode
This is really an easy usage version of Tournament mode. Instead of entering all four parameters needed for tournament mode, the program simply asks you 'Move time?' The program will then control its search to use on average the time you enter. (Entering 0 seconds for the average move time, selects the programs fastest level.) This effectively provides thousands of 'levels' of play.

Note 1: in fact, this mode sets up tournament parameters of 60 moves to each time control, with 60 times the average move-time being allocated for each control.

Note 2: this is the initial mode, with an average of 10 seconds per move.

3) All-the-moves mode
In chess clubs, most games are played with chess clocks, with each player having say five minutes to complete the game or 'lose on time'. This mode allows you to set the complete game-time and the program will try to play all the moves of the game within the specified time. If either player exceeds the game time, the game will be terminated with the message 'Time up'. By using the 'Elapsed-time clocks' command, you can 'handicap' the program or yourself, by putting some time on one of the clocks before starting the game. This can be very useful for players who are much worse or much better than the program.

4) Equality mode
In this mode, the program will attempt to keep its elapsed time clock, closely matched to your elapsed time clock, effectively playing at a similar speed as you are playing. If you make several quick moves, the program will also probably have to play quickly to keep up with the 'pace' you are setting. If however, you think for a long time over some moves, the program will also be able to think longer over its moves. In practice, the programs clock will usually trail yours by a few minutes.

5) Infinite mode
This mode can be used to solve chess puzzles of the form 'find the best move in this position'. It searches all combinations of all moves to ensure that it does not miss a sacrifice. It continues searching, further and further ahead, until you tell it to stop, or it finds a forced checkmate for either side, or it has searched 12 ply ahead (the programs maximum lookahead) or there is only one legal move in the position. This is very useful for postal chess games where you can just leave the program running overnight (or sometimes even longer) to find the best move. When the program starts searching for the move, the clocks are set to 00:00:00:, so that the time taken on the move can be recorded. Also note that the openings book is not searched in this mode. Of course, the program may find the same move in much less time using its normal playing mode, because certain 'obviously bad' moves are ignored.

6) Problem mode
This mode is used to solve chess mating problems. When selected the program asks 'Problem Type?'. The program can solve three types of problems:- 1) Normal mates - of the form 'White to move and mate black in X moves' 2) Self mates - of the form 'White to move and force black to mate white in X moves' and 3) Help mates - of the form 'White to move and help black to mate white in X moves'. Of course, Colossus can find mates for black just as easily. Colossus is the first home computer chess program ever to be able to solve selfmates and helpmates.
The program then asks 'Mate in?', whereupon you must enter the number of moves till the intended mate. Colossus can solve mates up to seven moves ahead. When the program starts seaching for the mate, the clocks are set to 00:00:00, so that the time to find the mate can be recorded. If the program finds a mating move, it prints the mating line below the move record, stops the clocks, and asks 'Continue?'. If you are satisfied with the discovered move, then pressing 'N' will cause the program to make this move immediately. If however you wish the program to continue searching for any alternative mate, press 'Y' and the program will restart the clocks and continue searching for another mate. This process is repeated until no more mating moves are found, in which case the program will then make the last mating move found, on the board. If the program does not find any mate, the message 'No mate' is displayed and the program automatically enters 'Alter-position' to allow you to select another position or correct the current one. Colossus is one of the fastest problem solving chess programs in the world!

Visible - Type <CTRL> 'V'
This allows you (if you feel confident enough) to play the equivalent of a game of 'blindfold' chess, as sometimes done by strong chess players. The program asks 'Visibility?' when you must enter a number from 0 to 3. (See section 4.b for details on how to enter numeric data.) The number entered determines which pieces on the board are made invisible. 0 is the default value where both sides pieces are displayed.

1 will make the white pieces invisible
2 will make the black pieces invisible
3 will make all the pieces invisible.

The moves are still indicated on the board and displayed in the move record. When 'Invisible' is selected, the word 'Invisible' appears at the top right of the board, to avoid confusion.

Write a printer - Type <CTRL>'W'
You can print the current board position, the current move-record or the moves as they happen (including the technical information if required) to a printer. The program asksa 'Board, Record or Moves?' If you type 'B' the current board position will be printed. The printout is always in the normal 'white-up-the-board' orientation as found in all chess literature. The white pieces are printed as capital letters, the black pieces as lowercase letters, the empty squares as dots. If you type 'R' the current move-record will be printed. A header is printed first for clarity. Also, if you have used 'Alter-position' then the initial position is printed before the list of moves. If you type 'M' the program responds 'Moves?'. You can then enter a value from 0 to 2 to select one of the three types of continuous move printout. A value of 0 gives no continuous printout (efault). A value of 1 prints moves by both sides as they are played. A value of 2 also prints the programs technical information for each of its moves. If you start a new game with continuous printout on, then the move-record header is printed for clarity. Any other keypress will abort the 'Write' command.
Note: the printer interface card is assumed to be in slot 1.

Zap beeper - Type <CTRL> 'Z'
The programs 'beeps' can be turned off/on with this command. When the sound is turned off, no further beeps will be heard. When the sound is turned back on a high-pitched beep is made to signify the sound is now back on.

SECTION 5 - PROGRAM INFORMATION

5.a - Statistics

Author: Mr. M. P. Bryant
Publisher: **CDS Software Ltd.,** Silver House, Silver St., Doncaster, England.
Language: 6502 assembly
Code size: 22K
Data size: 5K
Positions examined per second: 300 (average)
Estimated Rating: 1850+ ELO (156+ BCF)

5.b - Openings book

Colossus has an openings 'book' of about 3000 positions, which it uses to play the first few moves of the game quickly and accurately. Lines vary from 2 ply to 17 ply deep. When there is a choice of book moves at any position, Colossus will choose between them at random, with a slight bias towards the moves which Colossus itself considers best. The 'book is stored on the master disc and so if Colossus tries to examine the 'book', but you have removed the disc, then you will be prompted to re-insert it.

5.c - Results

Colossus 4.0 has been tested against numerous other home computer chess programs and has beaten them all. In the tests, sixteen games were played, on various levels, with an equal number of whites and blacks for each program. One point was awarded for a win, half a point for a draw, no points for a loss. The results of these tests were as follows:-

Colossus 4.0 (Apple II) beat the following programs by:-

Mychess 2.0	Beyond	CBM 64	10.6
White-Knight Mk12	BBC Publications	BBC	11.5
Sargon III	Hayden	Apple II	12.4
Superchess 3.5	CP Software	Spectrum	12.4
White-Knight Mk11	BBC Publications	BBC	14.2
Parker Chess	Parker Games	Atari	14.2
Cyrus IS Chess	Sinclair	Spectrum	14.2
Superchess 3.0	CP Software	Spectrum	15.1
Chess 7.0	Odesta	Apple II	16.0
Sargon II	Hayden	Apple II	16.0
Chess	Atari	Atari	16.0
Chess	Acornsoft	BBC	16.0
Chess	Bug-Byte	BBC	16.0
Chess	Computer Concepts	BBC	16.0
Chess	Program Power	BBC	16.0
Grandmaster Chess	Audiogenic	CBM 64	16.0
	Oasis	Dragon	16.0

Chess	Acornsoft	Electron	16.0
Chess	Program Power	Electron	16.0
Chess	Oric	Oric	16.0
Chess	Psion	Spectrum	16.0
Master Chess	Mikro-Gen	Spectrum	16.0
Spectrum Chess II	Artic	Spectrum	16.0
ZX Chess II	Artic	ZX81	16.0

Of course, the results cannot be guaranteed precisely, in every set of sixteen games. In practice they are bound to vary slightly either way, but they do give a very good guide to the relative strengths of the programs.

SECTION 6 - GAMES & PROBLEMS DISC VERSION ONLY

Saved on the master disc are a number of games and problems. These can be loaded with the 'Disc' command (described earlier). The games are a selection played by the best micro, mini and mainframe computer chess programs. You can attempt to solve the problems yourself or make the program show you the solution if you get stuck.

6.a. - Games

The final position of each game is stored, with the complete game record, which can be scanned with the 'Replay' command (described earlier).

File number	Game description
0	Computer v Human Friendly Simultaneous Game (Summer 1978):- Walter Browne v Chess 4.6 (CDC Cyber 176) [0-1 White resigned]
	Computer v Human Challenge Match (Autumn 1978):- (Included are two games of this match, which Levy won by 3.5-1.5)
1	David Levy v Chess 4.7 (CDC Cyber 176) [.5-.5 Agreed drawn]
2	Chess 4.7 (CDC Cyber 176) v David Levy [1-0 Black resigned]
	Computer v Human Friendly Game (Autumn 1980):-
3	Mike Valvo v Belle (Custom built computer) [0-1 White resigned]

Computer v Human Friendly Game (Autumn 1980):-

4 Belle (Custom built computer) v Hans Berliner [0-1 White resigned]

Computer v Human Friendly Game (March 1982):-

5 J.H. Donnor v Belle (Custom built computer) [1-0 Adj]

Computer v Human Friendly Simultaneous Game (September 1983):-

6 A. Karpov v Mephisto III (MC68000) [.5-.5]

Fourth World Computer Chess Championship (Autumn 1983):-
(Included are all the games by the winning program Cray Blitz and all the other first round games)

7 Cray Blitz (Clay 1 XMP) v BCP (Custom built computer) [1-0]

8 Fidelity X(6502 chess machine) v Cray Blitz (Cray 1 XMP) [0-1]

9 Cray Blitz (Cray 1 XMP) v Advance 3.0 (Custom built computer) [1-0]

10 Nuchess (Cray 1) v Cray Blitz (Cray 1 XMP) [.5-5]

11 Belle (Custom built computer) v Cray Blitz (Cray 1 XMP) [0-1]

12 Phoenix (Honeywell DPS 8170) v Belle (Custom built computer) [0-1]

13 Schach 2.7 (Burroughs B7800) v Chaos (Amdahl 5860) [.5-.5Adj]

14 Nuchess (Cray 1) v Philidor X(IBM PC) [1-0]

15 Merlin (Siemens 7.890) v Bebe (Chess Engine) [1-0]

16 Advance 3.0 (Custom built computer) v Sfinks X(TRS-80) [1-0]

17 Pion (Vax 11/750) v Fidelity X(6502 system) [0-1]

18 Mephisto X(68000 system) v Conchess X(6502 system) [1-0 Time]

19 Awit (Amdahl 5860/2) v Ostrich (Multicomputer system) [1-0]

20 Novag X(6502 system) v Bobby (IBM 4341-2) [.5-.5]

21	Shy(Burroughs B7800/B7900) v Patsoc 2.0 (DEC KL-10) [.5-5]
	Third World Micro-Computer Chess Championship (Autumn 1983):- (Included are all the games by the winning program Elite A/S)
22	Elite A/S v Mephisto Excalibur [1-0]
23	Logichess 2.2 v Elite A/S [0-1]
24	Elite A/S v Mephisto X [1-0]
25	Elite A/S v Novag X [.5-.5]
26	Chess 2001 X v Elite A/S [0-1]
27	Elite A/S v Super Constellation [.5-.5]
28	Superstar v Elite A/S [0-1 Time]
	Personal Computer Games magazine Home-Computer Chess Championship (Spring 1984):-
	(Included is one of the games from the final. Colossus 2.0 beat ten other home computer chess programs to win the tournament without losing a single game!)
29	Colossus 2.0 (CBM 64) v Superchess 3.0 (Spectrum) [1-0]
	Computer v Human Challenge Match (April 1984):-
	(Included are all the games of this match, which Levy won by 4-0)
30	Cray Blitz (Cray 1 XMP) v David Levy [0-1 White resigned]
31	David Levy v Cray Blitz (Cray 1 XMP) [1-0 Black resigned]
32	Cray Blitz (Cray 1 XMP) v David Levy [0-1 White resigned]
33	David Levy v Cray Blitz (Cray 1 XMP) [1.0 Black resigned]
	Computer v Human Friendly Game (April 1984):-
34	Victor Korchnoi v Nuchess (Cray 1) [1-0 Black resigned]

6.b - Problems

The initial position is stored, with the mating sequence, as found by Colossus, which can be scanned through by using the 'For-step' or 'Replay' commands (described earlier). White is to move unless otherwise specified.

File number	Problem description
100	Self-mate in 2
101	Self-mate in 2
102	Self-mate in 2
103	Self-mate in 3
104	Self-mate in 4
105	Self-mate in 5
106	mate in 2
107	mate in 2
108	mate in 3
109	mate in 3
110	mate in 3
111	mate in 4
112	mate in 4
113	mate in 4
114	mate in 4
115	mate in 5
116	Help-mate in 2 (Black to move)
117	Help-mate in 2
118	Help-mate in 2 (Black to move)

FIREBIRD LICENSEES INC. P.O. BOX 49, RAMSEY, N.J. 07446 HOTLINE: 201.934.7373